Gesture Based Control and EMG Decomposition

Kevin R. Wheeler, *Senior Member, IEEE,* Mindy H. Chang, and Kevin H. Knuth

(Invited Paper)

***Abstract*— This paper presents two probabilistic developments for use with Electromyograms (EMG). First described is a neuro-electric interface for virtual device control based on gesture recognition. The second development is a Bayesian method for decomposing EMG into individual motor unit action potentials. This more complex technique will then allow for higher resolution in separating muscle groups for gesture recognition.**

All examples presented rely upon sampling EMG data from a subject's forearm. The gesture based recognition uses pattern recognition software that has been trained to identify gestures from among a given set of gestures. The pattern recognition software consists of hidden Markov models which are used to recognize the gestures as they are being performed in real-time from moving averages of EMG. Two experiments were conducted to examine the feasibility of this interface technology. The first replicated a virtual joystick interface, and the second replicated a keyboard.

Moving averages of EMG do not provide easy distinction between fine muscle groups. To better distinguish between different fine motor skill muscle groups we present a Bayesian algorithm to separate surface EMG into representative motor unit action potentials. The algorithm is based upon differential Variable Component Analysis (dVCA) [1], [2] which was originally developed for Electroencephalograms. The algorithm uses a simple forward model representing a mixture of motor unit action potentials as seen across multiple channels. The parameters of this model are iteratively optimized for each component. Results are presented on both synthetic and experimental EMG data. The synthetic case has additive white noise and is compared with known components. The experimental EMG data was obtained using a custom linear electrode array designed for this study.

I. INTRODUCTION

Electromyograms (EMG) are used in the medical community to aid in the diagnosis of neuromuscular diseases, and there has been increasing interest in the use of EMGs as a means to interface with prosthetics and virtual devices [3]. For clinical applications it is often necessary to use invasive needle electrodes to pinpoint sources of the EMG to specific motor units. However, invasive measures are not ideal for use in the control of virtual devices. The ability to utilize surface EMG signals would enable the design of many neuro-electrically interfaced systems.

This paper introduces one such system in which surface EMG recordings from hand gestures are used in place of mechanical devices such as joysticks and keyboards to interface with a computer. Currently most gesture recognition systems come in one of two forms:

- Gestures are recognized via an external camera which requires sophisticated image processing and controlled lighting.
- Gestures are recognized by placing a sensing glove on the hand(s) of the participant.

We aim to achieve recognition in poor lighting conditions in extreme environments (outside of the lab) with minimal equipment. To date, we have accomplished this by directly connecting a person to the computer via EMG surface electrodes on the forearm. The EMG signals are sampled, digitized, and the resulting time-series are passed through a pattern recognition system based upon hidden Markov models (HMMs). The recognized patterns are then transmitted as computer commands. Our first example of this was to attach four pairs of electrodes to one forearm and interpret the resulting EMG signals as joystick commands [4]. These commands were then used to fly a realistic flight simulator for a 757 transport aircraft. The acting pilot would reach out into the air, grab an imaginary joystick, and then pretend to manipulate this stick to achieve left and right banks and up and down pitches of the aircraft simulation. We also present results on pretending to type on a table (or lap) and translating the resulting sensed EMG signals into keystrokes.

The demonstration of gesture recognition through surface EMG signals leads to physiological questions of how individual sources are involved in generating these EMG signals that are distinctive for different types of movements. Voluntary limb movement occurs as a result of the brain generating a spike train that is transmitted through the nerve to a junction in the muscle known as the endplate region. This induces an ion transfer along the length of the muscle fibers with a corresponding contraction of the muscles. The travelling waveform along the muscle fibers is known as a motor unit action potential (MUAP). This ion exchange induces a current on the surface of the skin which can be measured as a voltage via a resistive electrode. Surface EMGs measure a composite of the

K. Wheeler is the lead of Embedded Decision Systems in the Intelligent Systems Division at NASA Ames Research Center, Moffett Field, CA

M. Chang is a Ph.D. student in the Department of Bioengineering at Stanford University, Stanford, CA

K. Knuth is an assistant professor in the Department of Physics at the University at Albany, Albany, NY

voltage changes produced by these individual MUAPs. Measuring from the surface of the skin presents additional complexities because the multiple MUAP sources mix as they traverse through skin, fat, muscle, and other tissues.

In order to separate the EMG signals into the corresponding fine motor muscle groups we examine new ways to decompose EMGs. Thus, the unmixed MUAPs can be used as input to our hidden Markov models. We present a Bayesian method to perform source separation for surface EMGs. In particular, compound motor unit action potentials (CMAPs) [5] are separated into representative MUAP waveforms. Our method is based on the differentially Variable Component Analysis algorithm (dVCA) for source separation of Electroencephalograms (EEG) developed by Knuth et al. 2004 [1], [2]. We have extensively modified this algorithm to work with surface EMG.

In any standard Bayesian methodology, it is necessary to have a forward model and a means to optimize the parameterization of that model based upon data observations. In the case of EMG, we have chosen to develop a model that describes the MUAPs and how they are mixed together. There has been extensive research on decomposing EMG [6], [7], [8], [9] using non-Bayesian approaches. There has also been great progress made in developing physics-based forward models for EMG signal generation as measured on the surface of the skin [10], [11], [12], [13], [14]. Unfortunately, in most of this literature, there is a gap between the methods of decomposition and model parameterization that could be bridged by following a Bayesian approach. In this paper, we detail the steps that we have taken to fill this gap with a simple mixing model.

II. Methodology

A. Gesture Based Control

Each type of gesture set required a different methodology. The virtual joystick gesture set used four pairs of dry electrodes and four coarse grained movements. The virtual keyboard gesture set consisted of 8 pairs of wet electrodes and 11 fine grained movements. The methodology that we followed consisted of the following steps:

1) Gesture selection
2) Electrode application (location and number)
3) Signal acquisition, filtering, and digitization
4) Feature formation
5) Pattern recognition model training and testing
6) Pattern recognition application in interactive simulation

The process started by selecting the desired physical motions (gestures) to be used to control the virtual device. From the set of gestures, the best location for the limited number of electrode pairs (a maximum of 8 in our case) was established. Then standard signal processing practices were used to filter and digitize the signal. Transforms such as moving averages were applied to this raw digital data. The transformed data was fed into the pattern recognition software to train the models. Once the pattern recognition models were trained, they could be used for the real-time recognition task. Each of these steps will now be described in detail.

1) Gesture selection: Our first task used coarse grained gestures to mimic manipulation of a joystick [4]. Movement of the joystick was associated with four basic gestures: up, down, left, and right. The use of four pairs of electrodes for gesture recognition provided for reasonable separation between the four gestures.

Our second task consisted of movements associated with typing on a number pad on the keys 0 - 9 and Enter. These movements consisted of much finer grained gestures. The first, second, and third fingers were resting over the 4, 5, and 6 keys respectively. The first finger was used to press the keys 1, 4, and 7. The second finger was used to press the keys 2, 5, and 8. The third finger struck the keys 3, 6, and 9. The fourth finger was used for the Enter key, and the thumb was used to strike the zero key. In this case we used 8 pairs of electrodes.

2) Electrode application: The placement of the electrodes depends upon the gestures that we wish to recognize and upon individual physiological differences. The joystick task was measured using 4 dry electrode pairs sewn into a sleeve as shown in Figure 1. This sleeve helped to reduce variation in the placement of the electrodes. For the typing task, we chose to use eight pairs of wet electrodes due to the improved signal to noise characteristics of wet electrodes over that of dry electrodes. This was in part due to the signal amplitudes for the typing task being much smaller than that of the joystick task. The drawback to using wet electrodes is that the positions of the electrodes are difficult to replicate from one day to the next. The locations of these pairs were obtained by establishing a grid of electrodes on the forearm, and then performing the desired task; only those electrodes which produced distinct signals for a gestures were used. The positions of the electrodes for the typing task were in two rings around the forearm: one near the wrist, and one near the elbow also shown in Figure 1.

Several tests were conducted to measure the effects of minor variations in placement (1-3 mm) and major displacements (1-2 cm). The minor variations had no impact but the major displacements required that the recognition models be re-trained or adapted for the individual user. Individual differences in personal physiology proved to be challenging. Differences in arm lengths and widths made it difficult to place the electrodes at the proper positions across people without considerable effort. In addition, strengths of the EMG signals varied across people and varied with the amount of training that individuals received.

3) Signal acquisition, filtering, and digitizing: The EMG data was acquired by placing differential instru-

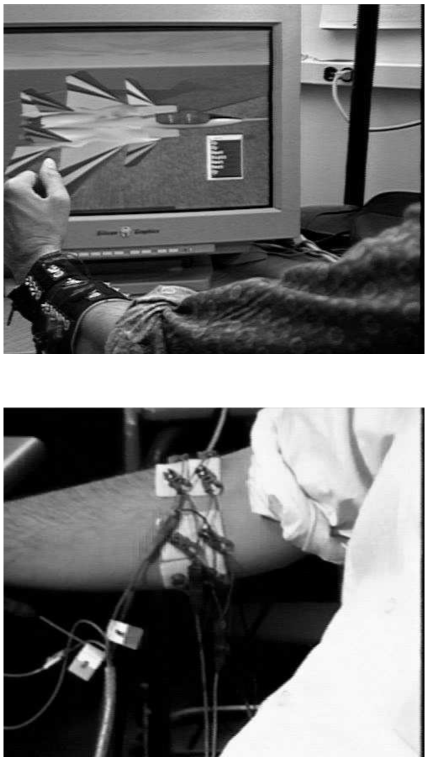

Fig. 1. Top: Dry Electrode sleeve for joystick based flying, bottom: wet electrodes for typing experiments

mentation pre-amplifiers near to each electrode pair with a Common Mode Rejection Ratio of 110 dB. All eight channel pairs were referenced to a common ground electrode positioned over bone at the wrist or elbow. The signal was digitized using 16 bits at 6000 Hz., and then a 32 tap anti-alias bandpass Bessel filter was applied and down sampled to 2000 Hz unless otherwise indicated.

4) Feature formation: The goal of the feature formation step is to separate the signals enough to allow the pattern recognition module to distinguish between gestures. Another result of working with features is to create a space smooth enough to be reliably modelled. We tried many common methods such as Short Time Fourier Transform (STFT), wavelets, moving averages, and auto-regression coefficients. In the end, moving averages, the simplest feature space, seemed to be the best. Since the EMG signals were differentially amplified, the average of the signals when presented with enough samples was approximately zero. This required that the moving average be performed on the absolute value of the signals. The windows used to form the moving averages were allowed to overlap by 75 percent. Note that this is purely an amplitude-based method; the frequency of the electrical activity did not seem to vary significantly from one gesture to the next.

5) Pattern recognition: The pattern recognition method we chose to employ was a hidden Markov model (HMM). HMMs have been developed by the speech recognition community in response to their pattern recognition time-series problem ([15]). The history of speech recognition reveals a process which first attempted to recognize isolated words from a single speaker, then isolated words from multiple speakers, followed by continuous words from a single speaker, and finally continuous words from multiple speakers. We are following a similar approach with our gesture recognition work. We have developed isolated gesture recognition for both a single participant and for multiple participants. The work described in this paper will describe isolated recognition for a single typist and continuous recognition for the joystick study.

Two issues with training any model to learn from sampled data are that the data set is representative and that the model has the appropriate number of parameters for accurate representation. The training data set can suffer from not having enough exemplars or being inconsistent for the sample size. In our case, we can always sample more data if we do not have enough. On an empirical basis, we have been able to use as few as 20 exemplars from each gesture to adequately model the remaining data from a single day. However, when we combine data from multiple days it becomes readily apparent that inconsistency is a problem.

We define inconsistency as the statistics of the data varying from day-to-day. We could have defined this in terms of gesture-to-gesture variation but have chosen not to because this variation is more of a natural variation inherent in human behavior whereas the day-to-day inconsistencies are more an artifact of the experimental procedures.

There are many solutions to resolving this inconsistency as well as many contributions to the variations which could be minimized. One example is electrode placement. If the electrode locations are allowed to vary from day-to-day then the signal statistics will also vary. This can be reduced through the use of a fixed electrode sleeve.

Day-to-day variations related to natural behavior may not be removable, and in fact we would benefit from modelling them. One example is the way which people gesture may vary slightly from day-to-day even though their intention is to perform the gestures identically. In this case we need to have enough data to represent the multi-modal statistics and we need a way to adapt the system models over time. Our current methodology does not vary adaptively but it is our plan to include this in future work. This means that our best remedy is to recognize when day-to-day variation is too great for adequate model generalization. We can then use less data for training by using only the data similar to our current day's setup (i.e. electrode locations).

a) Training: The HMMs we used were continuous, tied mixture [16], left to right models. Standard Baum-Welch training [15] was used. Models are classified as *continuous* if they use inputs which can take on a range of floating point values. The alternative to this is to allow for only discrete values such as might be found if the input were transformed by quantization. *Tied mixtures* means that a fixed number of Gaussian mixtures are used throughout all of the states. Thus any state may make use of any mixture. A *left to right* model means that the HMM may not go back to a previous state but may remain in a state or go to a new state.

Initialization of the models was performed using K-means clustering. The states were partitioned to equalize the amount of variance present within each state. The data sets used to train were segmented to insure that the peak of the variance was near the middle of each segment. This translated to the bulk of the energy being centered. Segments were sampled at 2000 Hz and contained 3072 samples per channel, with eight channels total. The parameters of the HMMs that we typically varied were the number of discrete states, the number of Gaussian mixtures, the number of maximum number of iterations to train, the method used to arrive at the state partitioning (uniform vs. variance based), and the method used to initialize the parameters of the mixtures (e.g. K-means clustering).

b) Recall: The real-time recall was performed using the standard Viterbi algorithm [17]. Since the system was processing streaming data, there was no knowledge as to where the peak of the variance was occurring. Because of this, the HMMs would see the data when the peak was first at the left most in the time segment, then the peak would move across from left to right, and then the final presentation was when the peak was at the right most part of the segment. Since the HMMs were trained only when the peak was centered, due to this shifting, the HMMs were required to recognize a gesture several times in a row before that gesture was selected as the one that was observed. This prevented spurious recognition when the peak was not near the center of observation.

6) Experiments: Two experiments were conducted in order to determine the feasibility of using bioelectric signals to substitute for, first, a joystick, and second, a keyboard.

The first experiment consisted of four pairs of dry electrodes fitted within a sleeve worn on the forearm of a participant. The participant was then asked to pretend to move a joystick left, right, up, and down. The participant performed each of these gestures 50 times. The data was separated by gesture, and segmented to have the peaks be in the center of 3072 sample segments. Artifacts or incomplete gestures were removed from the data sets via manual inspection. The segmented data were then used to train four HMMs, one for each gesture. These trained models were then used to recognize gestures made on a day excluded from the training set. A confusion matrix was generated to display errors and to show which gestures were confused with one another. The system has also been used for numerous real-time demonstrations of flying a simulated 757 transport aircraft to landing [4]. A more continuous gesture recognition was implemented by decreasing the segment size.

Four methods were used to test the pattern recognition system. The first involved training the models on data from one day, and then recalling on different data obtained on the same day. We call this method *same trial acquisition and testing*. The second involved training on data from one day and recalling on data collected on a different day. We call this method *cross-trial acquisition and testing*. The third method trained on data sub-sampled from a large set taken across multiple days, and then recall was performed on data different from the training but in the same large set. This third method we called *multi-trial acquisition and testing*. The final method involved training on a previously acquired single day that provided the best recognition in our real-time simulation for flying an aircraft. We call this *best trial training and real-time testing*.

The second experiment used eight pairs of wet electrodes in two rings of four each, one ring near the wrist, and the second near the elbow. The participant was asked to touch type on a printed picture of a number pad keyboard, striking the keys 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and Enter. The participant was asked to type these in order, separated by a one second rest interval, for a total of 40 strokes on each key. This data was then segmented, and artifacts were manually removed. Data were collected on several different days. Eleven HMMs were trained, one for each gesture. These eleven models were then run in parallel during recall.

The performance on batch data sets is not equivalent to the performance found in live demonstrations. Typically batch data sets are collected under static conditions. The live demonstrations are typically performed under high stress, with imperfect electrode placement, while the participant is bombarded with questions and distractions. Thus live performance tends to suffer from more errors than the batch testing results.

B. EMG Decomposition

The model that we formulate for separating mixed MUAPs is dependent upon how we acquire the data. Ideally, within a Bayesian framework we would model every part of the system. We would start by modeling the sources of the potentials and how the shape of the potentials is changed by transmission through the tissue. This would be followed by a model of the electrodes, the amplifier, and finally of the data acquisition card. The model we present relies on approximations to reduce the task of modelling all of these elements.

Our model is based upon the assumption that we can observe compound MUAPs along parallel fibers of a muscle group. This assumption is facilitated by using a linear electrode array [18] [10] as shown in Figure 2. We fabricated this electrode array with parallel silver bars spaced 5 mm apart. Figure 3 shows the data collected by this device on four differential channels (eight total silver bars). Note that a star has been placed over one of the action potential waveforms which is shifted between channels by an amount proportional to the conduction velocity. The muscle contraction under study was carefully controlled and can be assumed to be constant. The contraction level in this work is approximately 20 percent of maximum voluntary contraction.

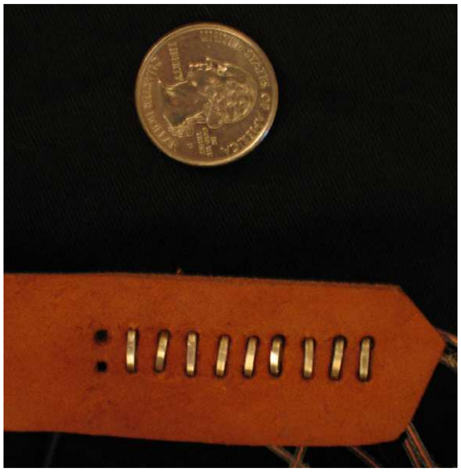

Fig. 2. Linear electrode array pictured with a U.S. quarter.

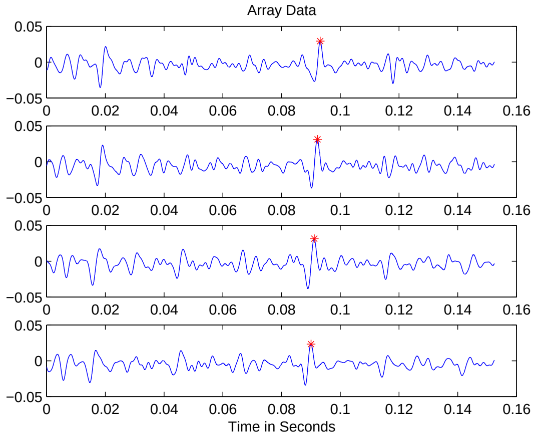

Fig. 3. EMG data from our linear electrode array. Star indicates moving MUAP over time between channels.

Our model representing the mixing process for the m^{th} channel as a function of time can be expressed as:

$$\psi_{m,t} = \sum_{n=1}^{N}\sum_{f=1}^{F} C_{mn}\alpha_{nf}s_n(t-(m-m_{\text{ref}})\tau_n^C - (f-1)\tau_n^F - \tau_{nf}^S) \quad (1)$$

where subscripts index the n^{th} component, f^{th} firing and m^{th} channel. N is the total number of MUAP sources (components) being modeled, F is the number of firings, C represents the coupling between channels and sources, $s_n()$ is the source waveform, and α_{nf} is the amplitude weighting. τ_n^F represents the time delay associated with the firing frequency of a particular source, τ_n^C is the delay across channels which is proportional to the conduction velocity, τ_{nf}^S is the latency for each source and firing representing the variability in firing.

There are several assumptions that underlie equation (1). We assume that the electrode array is positioned parallel to the muscle fibers and that the electrodes are evenly spaced. These assumptions allow us to model dominant components propagating along the muscle fibers as signals travelling from channel to channel. The time it takes to go from one channel to the next is represented by $\tau_n^C = \frac{d}{v_c}$. where $d = 5\text{mm}$ is the electrode spacing and v_c is the conduction velocity. We also assume that the muscle contraction is of constant force and that the sampling time is short enough that the firing rate (or time delay between firings τ_n^F) of any one MUAP source is effectively constant. Variation in the periodicity of the firing of a single source is modelled by τ_{nf}^S and is assumed small with respect to the firing rate.

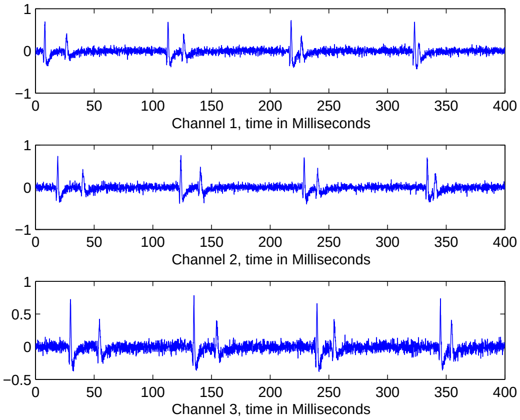

Fig. 4. Synthetically generated compound motor unit action potentials with two components and white noise added.

The basis of model parameter estimation lies in using Bayes' Theorem to maximize the a posteriori probability (MAP) of the model, using the likelihood of the data and the prior probability of the model parameters and other known information (symbolized by I):

$$p(\text{model}|\text{data, I}) = \frac{p(\text{data}|\text{model, I})p(\text{model}|\text{I})}{p(\text{data}|\text{I})} \quad (2)$$

Substituting the parameters of our model, this becomes

$$P = p(C, s(t), \alpha, \tau^F, \tau^C, \tau^S|x(t), I) = \frac{p(x(t)|C, s(t), \alpha, \tau^F, \tau^C, \tau^S, I)p(C, s(t), \alpha, \tau^F, \tau^C, \tau^S|I)}{p(x(t)|I)} \quad (3)$$

where the value on the left-hand side of the equation, which will be referred to as P, is the posterior probability of a model describing the data. The right side represents the product of the likelihood of data given the model and the prior probability of the model, divided by a proportionality constant dependent on the data. A uniform distribution is assigned to the prior probabilities of each parameter, and as a result the posterior probability P becomes directly proportional to the likelihood of the data:

$$P \propto p(x(t)|C, s(t), \alpha, \tau^F, \tau^C, \tau^S, I) \tag{4}$$

Using the principle of maximum entropy, the likelihood of the data is assigned a Gaussian distribution by introducing a new parameter σ. This parameter represents the expected squared error in prediction and is assigned a Jeffreys prior. When the likelihood is marginalized over all values of σ, the result becomes

$$P \propto (2\pi\sigma^2)^{-\frac{MT}{2}} \exp\left[-\frac{1}{2\sigma^2}Q\right] \tag{5}$$

where Q represents the square of the residuals between the data and our model, summed over all time points in all channels

$$Q = \sum_{m=1}^{M}\sum_{t=1}^{T}\Bigg(x_m(t) - \sum_{n=1}^{N}\sum_{f=1}^{F} C_{mn}\alpha_{nf}s_n(t- \\ (m - m_{ref})\tau_n^C - (f-1)\tau_n^F - \tau_{nf}^S)\Bigg)^2 \tag{6}$$

To simplify calculations we maximize P by maximizing the log of P. Using the method described by Knuth, et al. [1], [2], the log of the posterior probability P can be written as:

$$\ln P = -\frac{MT}{2}\ln Q + \text{const} \tag{7}$$

For convenience of discussion, two expressions frequently used in the process of minimizing the difference between the data and the model are defined below. For a given component j in channel m at time t, U represents all firings of the component j deduced from the value of the actual data minus all other parameterized components.

$$U(j,m,t) = x_m(t) - \sum_{\substack{n=1\\n\neq j}}^{N}\sum_{f=1}^{F} C_{mn}\alpha_{nf}s_n(t- \\ (m - m_{\text{ref}})\tau_n^C - (f-1)\tau_n^F - \tau_{nf}^S) \tag{8}$$

Similarly, the expression U_F isolates a particular firing, f_0 of the j^{th} component in channel m at time t, using the same method of deduction by also subtracting away all other firings of the j^{th} component except for the f_0^{th} firing.

$$U_F(j, f_0, m, t) = U(j,m,t) - \sum_{\substack{f=1\\f\neq f_o}}^{F} C_{mj}\alpha_{jf}s_j(t- \\ (m - m_{\text{ref}})\tau_j^C - (f-1)\tau_j^F - \tau_{jf}^S) \tag{9}$$

1) Parameters: The five parameters optimized through iteration are described below:

a) Waveshape: The Maximum A Posteriori estimate of the waveshape is found by setting the partial derivative of the log probability with respect to a time point q in waveshape s_j to zero. Details appear in Knuth 2005 [1].

b) Amplitude: When taking the partial derivative of the log probability with respect to the amplitude of the f_0^{th} firing of the j^{th} component, the optimal estimate for the amplitude of this particular firing becomes

$$\hat{\alpha}_{jf_0} = \frac{\sum\limits_{m=1}^{M}\sum\limits_{t=1}^{T} U_F R_\alpha}{\sum\limits_{m=1}^{M}\sum\limits_{t=1}^{T}(R_\alpha)^2} \tag{10}$$

$$R_\alpha = C_{mj}s_j(t-(m-m_{\text{ref}})\tau_j^C - (f_0-1)\tau_j^F - \tau_{jf_0}^S) \tag{11}$$

Since the model allows for varying amplitudes between different firings of the same component, each α_{jf} term is determined irrespective of other firings by using the deduced single firing term, U_F.

c) Firing Period: To find the optimal estimate for the firing period of the j^{th} component one must solve:

$$\hat{\tau}_j^F = \text{argmax} Y(\tau_j^F) \tag{12}$$

$$Y(\tau_j^F) = \sum_{m=1}^{M}\sum_{t=1}^{T} U(j,m,t)U(j,m,t+\tau_j^F) \tag{13}$$

where the function U is defined in equation (8). The function $Y(\tau_j^F)$ represents the autocorrelation across each channel, summed across all channels for all firings of a given component j. The j^{th} component is isolated by subtracting away all firings of all other components to obtain $U(j,m,t)$. Each channel is multiplied with shifted versions of itself, and assuming that the data is periodic across each channel, the latency shift that produces the maximal value will be where the 2^{nd} through F^{th} channel is closest to alignment with the 1^{st} through $(F-1)^{\text{st}}$ firing of the j^{th} component. This latency estimate is constrained to be positive and greater than 2 ms because a firing period that is significantly smaller than the time span of a single action potential is not physiologically plausible.

d) Conduction Period: To find the optimal estimate for the conducting period of the j^{th} component,

$$\hat{\tau}_j^C = argmax\ Z(\tau_j^C) \tag{14}$$

where

$$Z(\tau_j^C) = \sum_{m=1}^{M}\sum_{t=1}^{T} U(j,m-1,t)U(j,m,t+\tau_j^C) \tag{15}$$

where the function U is defined in equation (8). The function $Z(\tau_j^C)$ represents the cross-correlation between consecutive channels, summed across all pairs of channels for a given component j. As above, the estimate of the j^{th} component (with all of its firings) is obtained by subtracting away all firings of all other components to obtain $U(j,m,t)$. As a convention, the first channel will be considered the reference electrode from which action potentials are first detected. As action potentials propagate through the muscle fibers, each subsequent channel detects the action potential slightly later than the previous channel. For each pair of channels, the first channel is multiplied with shifted versions of the second channel. Assuming that the data is periodic across each channel, the latency shift between channels that produces the maximal value will be where all firings of the j^{th} component most nearly align between the pair of channels. The conduction period is constrained to positive values between zero and half of the firing period. Since the conduction period is significantly smaller than the firing period, we are able to apply this assumption.

e) Offset Latency: To find the optimal estimate for the offset latency of f_0^{th} firing of the j^{th} component,

$$\hat{\tau}^S_{jf_0} = argmax\ A(\tau^S_{jf_0}) \tag{16}$$

$$A(\tau^S_{jf_0}) = \sum_{m=1}^{M}\sum_{t=1}^{T} U_F(j,f_0,m,t)V_F(j,f_0,m,t+\tau^S_{jf_0}) \tag{17}$$

where the function U_F is defined in equation (9) and $V_F(j,f_0,m,t)$ represents the reconstruction of the f_0^{th} firing of the j^{th} component, using all other parameters of the j^{th} component:

$$\begin{aligned} V_F(j,f_0,m,t) \ &= C_{mj}\alpha_{jf_0}s_j(t-(m-m_{\text{ref}})\tau^C_j \\ &\qquad -(f_0-1)\tau^F_j - \tau^S_{jf_0}) \end{aligned} \tag{18}$$

The function $A(\tau^S_{jf_0})$ represents the cross-correlation between the deduced single firing of the j^{th} component, $U_F(j,f_0,m,t)$, based on the data after removing the firings of the other components, and the estimated single firing of the component, $V_F(j,f_0,m,t)$, based on the j^{th} component parameters. The deduced firing is multiplied with shifted versions of the reconstructed firing, and the latency which produces the maximal value is taken as the estimate of the offset latency.

2) Adjustments for Parameter Degeneracies:

a) Latency Degeneracy between τ^F_n and τ^S_{nf}: Since τ^F_n and τ^S_{nf} both represent time shifts within single channels of data, if the estimated value for τ^F_n is inaccurate, τ^S_{nf} values will increase linearly in amplitude. In other words, if the estimated τ^F_n value is smaller than the actual value, each successive firing will deviate from its estimate by a larger value than the previous firing with its respective estimate. For a given firing f of a given component n, the value of the net latency due to firing and offset is not affected, but τ^F_n and τ^S_{nf} values no longer represent the firing period and offset period. In order to correct for this offset, every time the τ^S_{nf} is calculated, a linear regression on the τ^S_{nf} values is performed and both latency values are adjusted accordingly. The linear regression takes the form of $\tau^S_{nf} = \mu_n f + \beta_n$ Using this line, τ^F_n and τ^S_{nf} values are remapped so that τ^S_{nf} becomes a constant value plus or minus deviations from the regression line, d_{nf}, and τ^F_n accounts for this change.

$$\begin{aligned} & (f-1)\tau^F_n + \tau^S_{nf} \\ = \ & (f-1)\tau^F_n + (\mu_n f + \beta_n + d_{nf}) \\ = \ & (f-1)\tau^F_n + (\mu_n(f-1) + \mu_n + \beta_n + d_{nf}) \\ = \ & (f-1)(\tau^F_n + \mu_n) + (\mu_n + \beta_n + d_{nf}) \end{aligned}$$

The adjusted τ^F_n value becomes

$$\bar{\tau}^F_n = \tau^F_n + \mu_n \tag{19}$$

and the τ^S_{nf} value becomes

$$\begin{aligned} \bar{\tau}^S_{nf} &= \mu_n + \beta_n + d_{nf} \\ &= \mu_n f + \beta_n + d_{nf} - (f-1)\mu_n \\ &= \tau^S_{nf} - (f-1)\mu_n \end{aligned} \tag{20}$$

b) Waveshape alignment and adjustment of τ^S_{nf}: A degeneracy also occurs in the time domain between the waveshape and the offset latency τ^S_{nf}. A time shift in the component could either be characterized as a change in waveshape or a shift in offset latency. In order to give the offset latency values a relative meaning between different components, each time a stopping condition is met, as in step 9 of the iteration process below, the peaks of all component waveshapes are aligned to match the waveshape with the earliest peak, and each τ^S_{nf} value is adjusted accordingly. This alignment is performed after each stopping condition to ensure that the algorithm has had a chance to estimate all parameter values before shifting all waveshapes to an earlier time. Performing the alignment during the iterative process runs the danger of shifting parts of the waveshape out of the time-frame allotted for a a single waveshape into negative time, which would be invalid for this model.

3) Iterations: We optimized the parameters as follows:

1) Identify the total number of firings within the dataset by human observation.
2) Estimate τ^F_n for the component using (12), (13).
3) Estimate τ^C_n using (14), (15).
4) Estimate τ^S_{nf} using (16), (17).
5) Adjust τ^F_n values if τ^S_{nf} values show a linear rate of change using (19),(20).
6) Estimate the waveshape $s()$
7) Estimate the amplitude α of each firing using (10).
8) To parameterize another component, follow steps 1-5, using the data from which the model of the first component has been subtracted.
9) Iterate through steps 1-5 for both components until the average change in waveshapes from the previous iteration is less than 1% or until a maximum number of iterations has been performed, and align the peaks of the component waveshapes, adjusting τ^S_{nf} accordingly.
10) For each additional source, parameterize the new component based on the data without all other components that are already modeled, and repeat

the iteration of steps 1-5 for all components until a stopping condition in step 9 is reached.

Listed below are considerations used in determining the above iteration order.

a) Parameter Initialization: Since the general waveshape of a MUAP is fairly well-defined, this information is used to initiate the waveshape. The point values in $s(t)$ are determined in the manner described below in the Synthetic Data section. As mentioned earlier, the coupling matrix is set to all ones under the assumption that all detectors receive signals from all components equally well. All latency values, τ_n^F, τ_n^C,and τ_{nf}^S are initialized as zero, and all α_{nf} values are one. Since the latency optimizations occur first in the iteration in steps 2-4, α_{nf} values are initialized as one. If the α_{nf} values were zero, the reconstructed component used in the τ_{nf}^S calculation would just be a straight line at zero. For this reason, if any value of α_{nf} becomes zero in the process of iteration, the α_{nf} value is set to one temporarily for the calculations of τ_n^F, τ_n^C, and τ_{nf}^S and then set back to zero.

b) Estimation of τ_j^F and τ_j^C before τ_{jf}^S: When considering a single component, τ_j^F and τ_j^C are parameters that characterize the component, providing information about the firing rate and conduction velocity when the distance between electrodes is known. The optimizations of τ_j^F and τ_j^C involve correlations of the deduced component and are not directly dependent upon the accuracy of the parameters of the component in question. On the other hand, τ_{jf}^S "picks up the slack" in the overall latency value and is restrained to be a constant with slight deviations for each firing. Effectively, this constant gives information about the relative offset between different components, and the deviation represents the time error between the model and actual data for each firing of this particular component. τ_{jf}^F involves the cross-correlation of the deduced component and the reconstructed component and therefore is directly dependent upon the accuracy of the j^{th} component parameters.

In determining the order of this algorithm, the τ_{jf}^S calculation is performed after τ_j^F and τ_j^C so the offset calculation has the benefit of using the already parameterized firing and conduction period values when reconstructing the j^{th} component for cross-correlation. When the latency values are remapped in step 5, the firing period adjustment is applied to a τ_j^F value that represents an estimate of the firing period rather than an initial value with no significance.

c) Estimation of the waveshape before amplitude: Due to the degeneracy that could occur in the model between the scaling of the waveshape and the amplitude, the waveshape is constrained to have a peak-to-peak amplitude of one to give the α values a consistent meaning. Thus in each set of iterations, the waveshape is estimated first and scaled peak-to-peak. The amplitude α is parameterized after the waveshape has been determined so that α can appropriately compensate for the waveshape scaling in each firing of the component in question.

d) Isolated optimization of new component on its first iteration: When parameterizing a new component j, all previous components have already been optimized to their stopping condition. In all parameter optimization calculations (steps 1-5), the deduced component is used, either in the form of a single firing or an action potential train. For the first iteration, since the j^{th} component has not yet been completely parameterized, using this component in calculations for other components may throw off parameter values unnecessarily. Therefore, for the first iteration of a new component j, all parameters of j are optimized. For successive iterations, all parameters are optimized for each component in turn.

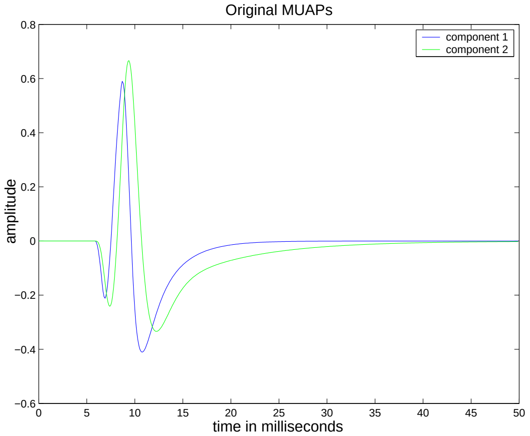

Fig. 5. Synthetic motor unit action potentials

4) Synthetic Data: The waveshape of the synthetic data used for testing is based on the MUAP model developed by McGill, Lateva, and Xiao 2001 [19]. The source function, V' (t) was created by the sum of a scaled spike and afterpotential:

$$\frac{dV(t)}{dt} = (ag'(t) + bg(t)) - \frac{b}{t_A} g(t) * \left(e^{-t/t_A} u(t)\right) \quad (21)$$

where * denotes a convolution, a and b are scaling factors, t_A is a time constant of decay, and

$$g(t) = \frac{k^{n+1}}{\Gamma(n)} t^n e^{-kt} u(t) \quad (22)$$

in which n and k are adjustable constants. The standard values used were $n = 2.5$ and $k = 5.8$ [19]. This source function was used as the initial estimate of all new component waveshapes. The model described by McGill, et al. also details spatial and temporal weighting functions to be convolved with the source function representing the waveshape distortion as it travels along the muscle fibers, as well as considerations for different lengths of muscle fibers. These factors were not implemented for this paper, but a convolution was performed using an approximate weighting function in generating the synthetic data. For simulation, varying levels of Gaussian noise were added as well. The data shown in Figure 4 was generated with

TABLE I
CONFUSION MATRIX FOR CROSS-TRIAL JOYSTICK DATA

Gesture	Left	Right	Up	Down	Correct
Left	15	0	26	9	30%
Right	0	50	0	0	100%
Up	0	0	50	0	100%
Down	0	0	1	49	98%

TABLE II
CONFUSION MATRIX FOR CROSS-TRIAL SHORT JOYSTICK DATA

Gesture	Left	Right	Up	Down	Correct
Left	17	0	25	8	34%
Right	0	49	0	1	98%
Up	0	0	50	0	100%
Down	0	0	0	50	100%

the two components shown in Figure 5, that were mixed together with uncorrelated white noise with a signal to noise ratio of 3.7. This level of noise is higher than that normally observed in our experimental setting, and thus is representative of a more difficult test.

5) Experimental Data: The subject EMG data was acquired using our electrode array positioned over the bicep. The subject was required to lift and hold a 5 pound weight and was only allowed to bend at the elbow, with the elbow supported. The data was sampled at 32 kHz. using a custom built amplifier with a gain of 1000 and an anti-aliasing filter with a 3 kHz cutoff frequency.

III. RESULTS

A. Joystick Gestures

1) Same trial acquisition and testing: This experiment is by far the easiest to recognize because the variability associated with day-to-day differences has been eliminated. Such variation includes conductivity levels of the skin, positioning of the dry electrode sleeve, and changes in the performance of the gestures. We noticed that we could determine when participants had used skin moisturizer before the experiment because the signal quality obtained from the dry electrodes improved. Typically if the HMMs had an appropriate number of parameters and enough data were used, then no errors were made upon recall of the validation set.

2) Cross-trial acquisition and testing: This experiment demonstrated which gesture was the hardest to separate from the others. In particular, we trained on 50 instantiations from one trial date and then validated on 50 other instantiations from a different trial date. A typical confusion matrix is shown in Table I. This indicates that day-to-day variations were significant enough to cause difficulty in separating the gesture Left from other gestures. The source of the variations included electrode placement, length of the gesture, strength of gesture formation, and the form of the gesture (wrist angles). This led to the next experiment to see if the models would generalize if we trained on all of the different days together and then tested on a withheld subset.

3) Multi-trial acquisition and testing: To determine the generalization capability of the HMMs we trained on data from multiple trial dates and then recalled on points withheld from the training data for the same dates. This resulted in perfect results (100% correct for all gestures).

Of course this does not mean that when yet another new day of data is added that the system will be able to generalize on that data. In the next experiment, we examine training on the single best day and use that for real-time testing. The real-time testing acts as new and unseen data.

4) Best-trial training and real-time testing: The error rates determined from the previous methods were not necessarily indicative of real-time performance. In particular, the error rates would vary across time depending upon many factors such as sleeve position (rotation), sweating, skin moisture (dry skin does not conduct well), length of time that the electrodes were worn, and fatigue (resulting in tremors). By training on only a single day's data, we were able to use the dry electrode sleeve for demonstrations on many different days. We selected the day which gave the best real-time reliability. The demonstrations consisted of flying a 757 transport aircraft in simulation to landing at San Francisco airport.

5) Continuous Recognition: In the previous experiments, a total of 3072 data samples were used to form the estimate. This introduced considerable time lag into the system (1.5 seconds). In an attempt to become closer to a continuous recognition process, the time segments used to train the HMMs were shortened to only contain the first part of the rise of the signal using 352 samples. In this case the HMMs consisted of 3 states with 9 mixtures total. The resulting cross-trial confusion matrix is shown in Table II.

This matrix is not significantly different from the previous cross-trial longer data. The change in signal length allowed for us to remove noticeable delays between the gesture action and the movement of the aircraft. We achieved a much less noticeable delays (176 ms.) at the expense of a slight decrease in the robustness of the gesture recognition process. It is possible to halve this response time with a small decrease in the recognition rates. Since this is intended for real-time systems, such a lag is hard to justify but it has not prevented us from successfully flying the simulated aircraft. The resulting multi-day confusion matrix had no errors.

In the next set of experiments we switched from using the dry electrode sleeve to wet electrodes. This was necessary because the EMG signals measured for the typing gestures were much smaller than those for the joystick. The wet electrodes tend to have a higher signal-to-noise ratio than the dry electrodes.

TABLE III
MULTI-TRIAL CONFUSION MATRIX FOR TYPING DATA

	1	2	3	4	5	6	7	8	9	%
1	46	0	0	4	0	0	1	0	0	90
2	0	48	0	0	0	0	0	3	0	94
3	0	0	49	0	0	1	1	0	0	96
4	11	0	0	38	2	0	0	0	0	75
5	1	3	0	5	36	1	3	2	0	71
6	0	1	6	0	0	42	0	0	2	82
7	0	0	0	0	0	0	51	0	0	100
8	0	0	0	0	2	1	3	44	1	86
9	0	0	0	0	0	0	0	0	51	100

B. Keyboard

In this experiment, the position of the hand above the simulated number pad was maintained in a touch-typist typing position. If the position of the hand were allowed to vary, the tasks of distinguishing between hitting the top row of keys from the bottom row of keys would greatly increase in difficulty and would require electrodes on the upper arm to sense the movement. The angle of the participant's wrist also had to be carefully maintained to avoid radically changing the sampled signals. Even with careful attention to position and maintaining electrode placement from day-to-day, the data tended to vary. There are consistent trends in the data but the variation between instantiations of the same gesture is great.

1) Multi-trial acquisition and testing: The keyboard replication experiments had much greater daily variation in electrode placement than with the joystick. We also had difficulty in reliably having the participants maintain a consistent hand position from trial to trial. This included making sure that the wrist angle was similar and that the hand was consistently neither resting on the table during motion nor was in part supported by the table (i.e. bad form, but consistent bad form). Despite these uncontrollable variations, the resulting confusion matrix for multiple trials shown in Table III looks pretty good. The variability in the data caused our models to generalize gestures such that more confusion occurred. For live demonstrations, we needed to train on the same day that we were giving the demonstration, and thus used only a single day's data.

Two enhancements are planned. First, the use of wet electrodes caused unintentional misplacement. We are currently developing new dry electrode straps which have a higher density than the sleeve and which are similar in size to a wrist band. These straps will allow us to have the electrodes on a band positioned relative to each other without variability. The second enhancement is to include model correcting adaptation which is now common in the speech recognition community. This adaptation would allow the models to be tuned to small variations, both throughout the day and whenever the current day's configuration differs from the models used to train. A calibration stage will be included so that the participant can make a gesture to issue a certain command and the computer will adapt to understand the signals as that command. Calibration will eliminate the need to require a participant to learn a fixed set of gestures. Instead, the person will be able to perform a gesture that seems natural to him or her in order to accomplish a given task, and the computer will simply map those signals to the correct action.

C. MUAP Analysis

1) Synthetic data: We applied the algorithm to 20 trials of synthetic data (see 4). The data consisted of three channels with only four firings across the channels. A typical decomposition is shown in Figure 6. This resulted in a median RMS error of 0.0297.

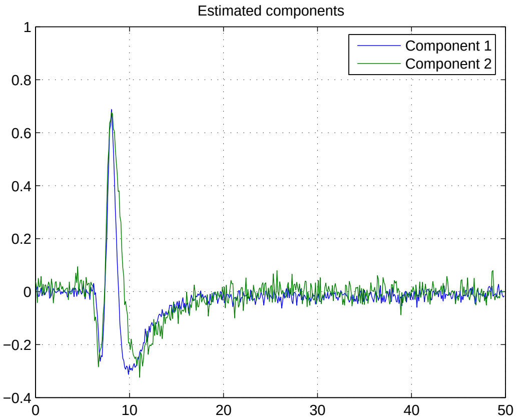

Fig. 6. Recovered components from synthetic data

2) Experimental Data: The experimental data, which is partially shown in Figure 7, was decomposed into two components shown in Figure 8. Since this was real data we do not have the actual MUAPs to which to compare our decomposition, so instead we compare this with a method in which clean MUAPs were hand-picked and then averaged together. This averaged MUAP waveform is depicted in Figure 9. which illustrates that there is qualitative agreement between the expected MUAP and the first component discovered using this algorithm. The second component contains multiple compound MUAPs because only two components were specified to be calculated.

IV. CONCLUSION

We have shown that it is feasible to control virtual devices via non-invasive EMG signal monitoring without explicitly modeling the EMG signals. Using hidden Markov models for pattern recognition, we have demonstrated the the ability to replicate movements associated with both joysticks and keyboards. We chose to demonstrate these input devices because they are familiar to the general computer user. Ultimately, improved interfaces will consist of more natural movements than those associated with either joysticks or keyboards. When our on-line adaptation software has been completed, a person

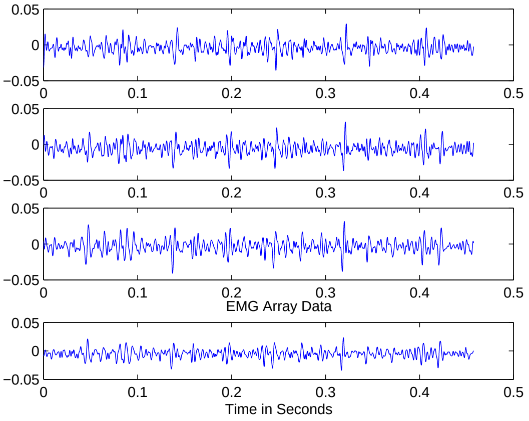

Fig. 7. Experimental EMG data.

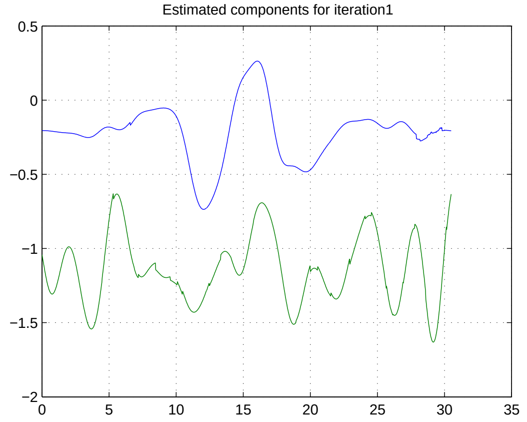

Fig. 8. Two components separated from experimental data.

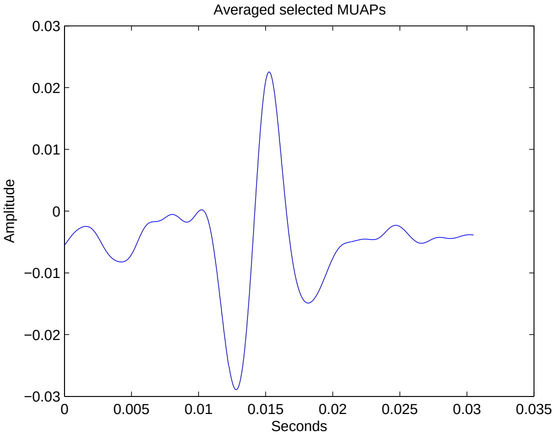

Fig. 9. Average waveform of hand selected MUAPs.

will be able to make the gesture he feels is natural for a given task (assuming that we have enough electrodes to cover the muscles involved) and the computer will map from this “natural” signal space to the expected “computer command” space.

Although the dry electrode sleeve guaranteed that the electrodes would be positioned consistently relative to each other, there was no guarantee that the sleeve would be in the exact same location on the arm. It was our hope that the sleeve would minimize day-to-day variations, but it needs to be redesigned to assure positioning. We would also like to increase the number of electrodes to allow for spatial oversampling. The dry electrode sleeve can suffer from intermittent conductivity problems when the impedance between a dry electrode and the skin become temporarily elevated due to hairs lifting the sensors or variations in the moisture level of the skin. In this study, the decrease in conductivity could usually be identified and fixed by manual readjustment of the sensors, but this inconsistency is not acceptable for daily use.

Ultimately, we envision a variety of applications for this work. The ability to naturally interface with a computer allows for humans to manipulate any electrically controlled mechanical system. In addition to wearable computing applications, we are also examining interfaces to robotic arms, mobile robots for urban rescue, unmanned aircraft drones, robotic exoskeletons, and space suit interfaces. There are also side benefits to using EMG signals for control in long duration space missions. One of the side effects of living in a zero gravity environment for extended periods is muscle atrophy. It would be possible to have astronauts train during a long flight to a distant planet by simulating the motions necessary to accomplish a given task. The EMG signals generated from these motions could be analyzed. If significant variations were detected, the astronaut could be given advanced warning to change their training routines to minimize atrophy and ensure mission success.

The drawback to using moving averages of EMG as input to the HMMs for gesture recognition was that the individual sources of EMG could no longer be distinguished. In order to distinguish smaller muscle group activations and inturn improve recognition, we have presented a dVCA algorithm for EMG decomposition based on Bayesian methodology. The original dVCA algorithm designed for EEG was substantially modified for the purpose of separating compound MUAPs measured in surface EMG. This modified algorithm was demonstrated on both simulated and real EMG data. The results are encouraging for both the synthetic and real cases. The most flexible part of this algorithm is that it allows the waveform $s(t)$ vary over time. Letting the waveshape be pointwise estimated permits it to have any shape, even shapes which are not at all physiologically plausible. This flexibility was deliberate to determine whether the algorithm would discover waveshapes resembling expected MUAP shapes. Indeed, we were pleased to see that the discovered components do resemble the synthesized MUAPs. We have obtained MUAPs from surface electrodes that are remarkably similar to our expected waveform without imposing any knowledge of what we were expecting to see in terms of shape.

The EMG decomposition method shown is a simple forward mixing model, which could eventually be replaced with a much more complicated physics-based model such as the electro-magnetic model of [12]. This would then allow for the automated determination of the representative tissue properties via proper model parameterization at the expense of more complex optimization.

Acknowledgements: The authors wish to thank the NASA CICT/ITSR program for supporting this research under the Neuro-electric Machine Control project, and the NASA IS/IDU program for support of the initial algorithm development.

References

[1] K.H. Knuth, A.S. Shah, W.A. Truccolo, S.L. Bressler, M. Ding, and C.E. Schroeder. Differentially variable component analysis (dvca): Identifying multiple evoked components using trial-to-trial variability. Submitted, preprint at: www.huginn.com/knuth, 2005.

[2] K. H. Knuth, W.A. Truccolo, S.L.Bressler, and M. Ding. Separation of multiple evoked responses using differential amplitude and latency variability. In T.W. Lee, T.P. Jung, S. Makeig, and T.J. Sejnowski, editors, *Proceedings of the Third International Workshop on Independent Component Analysis and Blind signal Separation: ICA 2001*, 2001.

[3] Kevin R. Wheeler and Charles C. Jorgensen. Gestures as input: Neuroelectric joysticks and keyboards. *IEEE Pervasive Computing*, 2(2), April-June 2003.

[4] Charles Jorgensen, Kevin Wheeler, and Slawomir Stepniewski. Bioelectric flight control of a 757 class high fidelity aircraft simulator. In *Proceedings of the World Automation Congress*, 2000.

[5] Robert Plonsey and David G. Fleming. *Bioelectric Phenomena*. McGraw-Hill Book Company, New York, 1969.

[6] Kevin C. McGill. Optimal resolution of superimposed action potentials. *IEEE Transactions on Biomedical Engineering*, 49(7), July 2002.

[7] Daniel William Stashuk. Decomposition and quantitative analysis of clinical electromyographic signals. *Medical Engineering & Physics*, 21, 1999.

[8] Daniel Zennaro, Peter Wellig, Volker M. Koch, George S. Moschytz, and Thomas Laubli. A software package for the decomposiion of long-term multichannel emg signals using wavelet coefficients. *IEEE Transactions on Biomedical Engineering*, 50(1):58–69, January 2003.

[9] Jianjun Fang, Gyan C. Agarwal, and Bhagwan T. Shahani. Decomposition of multiunit electromyographic signals. *IEEE Transactions on Biomedical Engineering*, 46(6):685 – 697, June 1999.

[10] Dario Farina, Elena Fortunato, and Roberto Merletti. Noninvasive estimation of motor unit conduction velocity distribution using linear electrode arrays. *IEEE Transactions on Biomedical Engineering*, 47(3):380 – 388, March 2000.

[11] Dario Farina and Roberto Merletti. A novel approach for precise simulation of the emg signal detected by surface electrodes. *IEEE Transactions on Biomedical Engineering*, 48(6):637–646, June 2001.

[12] Dario Farina, Luca Mesin, Simone Martina, and Roberto Merletti. A surface emg generation model with multilayer cylindrical description of the volume conductor. *IEEE Transactions on Biomedical Engineering*, 51(3):415–426, March 2004.

[13] Roberto Merletti, Loredana Lo Conte, Elena Avignone, and Piero Guglielminotti. Modeling of surface myoelectric signals - part i: Model implementation. *IEEE Transactions on Biomedical Engineering*, 46(7):810–820, July 1999.

[14] Roberto Merletti, Serge H. Roy, Edward Kupa, Silvestro Roatta, and Angelo Granata. Modeling of surface myoelectric signals - part ii: Model-based signal interpretation. *IEEE Transactions on Biomedical Engineering*, 46(7):821–829, July 1999.

[15] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257 – 286, February 1989.

[16] Jerome R. Bellegarda and David Nahamoo. Tied mixture continuous parameter modeling for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(12):2033 – 2045, December 1990.

[17] G. Daivd Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268 – 278, March 1973.

[18] Tadashi Masuda and Tsugutake Sadoyama. Topographical map of innervation zones within single motor units measured with a grid surface electrode. *IEEE Transactions on Biomedical Engineering*, 35(8):623 – 628, August 1988.

[19] Kevin C. McGill, Zoia C. Lateva, and Shaojun Xiao. A model of the muscle action potential for describing the leading edge, terminal wave, and slow afterwave. *IEEE Transactions on Biomedical Engineering*, 48(12):1357 – 1365, December 2001.

Kevin R. Wheeler received his B.S. (1988) and M.S. (1991) in electrical engineering from the University of New Hampshire, and his Ph.D. in electrical engineering from the University of Cincinnati (1996) specializing in statistical signal processing. After graduation he joined I.B.M. Almaden Research center to develop web-mining algorithms. He left I.B.M. to work at the Computational Sciences division at NASA Ames Research Center (ARC) in Moffett Field C.A. in 1997. He is currently group lead of Embedded Decision Systems in the Intelligent Systems Division at NASA ARC. His research group is focused upon applying probability theory to automating problems in Earth science, planetary exploration, man-machine interfaces and dexterous robotics.

Mindy Chang received the B.S. degree in electrical engineering and computer science with a minor in biomedical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, MA in 2005. She is currently a graduate student at Stanford University, Stanford, CA working toward the M.S. and Ph.D. degrees in bioengineering. Her research interests include signal processing and modeling of neural systems.

Kevin H. Knuth received his B.S. from the University of Wisconsin-Oshkosh (1988), his M.S. in physics from Montana State University (1990), and his Ph.D. in physics with a minor in mathematics at the University of Minnesota (1995). He held postdoctoral positions studying neuroscience at Louisiana State University Medical Center (1996), the City University of New York and the Albert Einstein College of Medicine (1997-1998), where he was later an instructor. He was also an instructor at the Weill Medical College of Cornell University (1999) where he worked on neurodatabases, and was a research scientist at the Center for Advanced Brain Imaging at the Nathan Kline Institute (1999-2001). From 2001-2005 he was a research scientist at NASA Ames Research Center where he developed machine learning techniques and their applications in the Intelligent Systems Division. In the Fall of 2005 he accepted an Assistant Professorship in the Physics Department and the College of Computing and Information at the University at Albany (SUNY).